

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



# THESIS

THE DESIGN OF A CELESTIAL NAVIGATION  
MICROCOMPUTER WITH THOUGHTS ON AN INTEGRATED  
INFORMATION DISTRIBUTION SYSTEM

by

Jon Prichet Moore  
David Brandt Rainsberger

June 1975

Thesis Advisor:

V. M. Powers

Approved for public release; distribution unlimited.

Thesis  
M802

DUDEY KNOX LIBRARY  
NAVAL POSTGRADUATE SCHOOL  
MONTEREY, CALIFORNIA 93944

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



# THESIS

THE DESIGN OF A CELESTIAL NAVIGATION  
MICROCOMPUTER WITH THOUGHTS ON AN INTEGRATED  
INFORMATION DISTRIBUTION SYSTEM

by

Jon Prichet Moore

David Brandt Rainsberger

June 1975

Thesis Advisor:

V. M. Powers

Approved for public release; distribution unlimited.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered):

of this nature brings to light the need for an information distribution system. Thoughts on a system of this purpose are presented.

DD Form 1473  
1 Jan 73  
S/N 0102-014-6601

UNCLASSIFIED  
SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



The Design of a Celestial Navigation  
Microcomputer with Thoughts on an Integrated  
Information Distribution System

by

Jon Prichet Moore  
Lieutenant, United States Navy  
B.S., Oklahoma State University, 1968

and

David Brandt Rainsberger  
Lieutenant, United States Navy  
B.S., Miami University, 1968

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE





## ABSTRACT

This report lays the groundwork for further development of a microcomputer that provides full celestial navigation capability. The physical design specifications and design philosophy are investigated. The fix algorithm which allows determination of position without the use of H.O. Publications is implemented in the BASIC language.

The application of microcomputer technology to devices of this nature brings to light the need for an information distribution system. Thoughts on a system of this purpose are presented.



## TABLE OF CONTENTS

I.	INTRODUCTION - - - - -	6
II.	NEED FOR A CELESTIAL MICROCOMPUTER PERIPHERAL- - -	8
III.	DESIGN PHILOSOPHY- - - - -	11
	A. OBJECTIVES - - - - -	11
	B. PHYSICAL DESIGN SPECIFICS- - - - -	12
IV.	THE CELESTIAL FIX ALGORITHM- - - - -	14
V.	PROGRAM THEORY AND IMPLEMENTATION- - - - -	22
	A. THEORY - - - - -	22
	B. IMPLEMENTATION - - - - -	31
VI.	THE INFORMATION DISTRIBUTION SYSTEM- - - - -	36
	A. INFORMATION DISTRIBUTION - - - - -	36
	B. DESIGN PHILOSOPHY- - - - -	38
VII.	LANGUAGE SELECTION - - - - -	43
VIII.	CONCLUSIONS- - - - -	48
APPENDIX A:	CELESTIAL FIX PROGRAM LISTING- - - - -	51
APPENDIX B:	SAMPLE PROGRAM RUN - - - - -	70
BIBLIOGRAPHY-	- - - - -	72
INITIAL DISTRIBUTION LIST	- - - - -	74
FORM DD 1473-	- - - - -	



# LIST OF FIGURES

Figure	Title	Page
1	Celestial Navigation Microcomputer - - - - -	13
2	Geographic Position (GP) of a Star - - - - -	15
3	Navigational Fix from Altitudes of Two Celestial Bodies - - - - -	18
4	Triangles Used for Navigational Fix- - - - -	19
5	Overall Celestial Situation with Six Intersection Points Depicted - - - - -	21
6	Information Distribution System- - - - -	41



## I. INTRODUCTION

This paper deals with two interleaved subjects. The two subjects will be discussed with varying levels of detail and scope. The more specific subject of the two is the conceptual design of a celestial navigation computer. A general approach toward a digital information distribution system is the second subject. The celestial navigation computer is discussed first, as it is an element of the entire system, and the design specifics of this device relate better to the entire system than vice versa.

The celestial navigation computer provides for manual input of information as well as documentation of the output for legal records. The main reason why a computer is beneficial for celestial positioning is that it eliminates errors made in recording, computing and plotting of celestial information. The design efforts attempt to eliminate manual recording of substantive information and manual plotting of lines of position (LOP's) for use in determining position. Automatic information recording is provided.

This paper provides an analysis of the programming language, program I/O requirements and physiological requirements of the device. The initial purpose of this paper was to design the device, then to see how this philosophy and design process could be applied to other building block devices applicable to the information distribution system. Comments to this point will be made in Section VI.





## II. NEED FOR A CELESTIAL MICROCOMPUTER PERIPHERAL

It would seem that with all of the fancy new gadgetry on the market today, the age-old method of navigating by the stars might just start to fade away. It is conceivable that this could happen in the civilian world; however, the military is often more demanding in the constraints placed upon its ships. The electronic devices which use electromagnetic radiation in the position determination loop are subject to deception, jamming and detection (as in the case of radar), and support failure, e.g., destruction of navigational satellites or LORAN stations, etc. The only other navigation device not subject to standoff disabling is the ship's inertial navigation system. It must be pointed out that an inertial navigation system is only as good as its platform's stability and the stability only as good as its positional estimates. Thus the longer the platform goes without being updated, the worse the platform stability becomes--truly a spiraling decline in accuracy.

With all of the electronic fixing devices out of the picture, visual sighting or sonar positioning is all that is left. Visual sighting is passive and includes celestial position determination, and this is the reason for maintaining excellence in position determination by the stars.

The microcomputer is a newcomer to this field. This technology properly applied allows for increased speed and



accuracy and the reduction of human errors. A need to determine position faster simply goes along with the increase in ship speed, e.g., new hydrofoil ships can conceivably reach speeds in excess of 60 knots. The need for increased accuracy is an economic as well as operational benefit. Accurate positioning is needed for proper and efficient deployment of weapons, efficient navigation and maneuvering.

The submarine force has implemented a celestial navigation system in the most complete terms. Completely computer controlled, several "spot shots" are obtained in a moment and the submarine's navigation systems can then be updated. The Air Force has used the ASTRO-TRACKER to aid in celestial navigation.

In January, 1975, Carl Nuese and Dr. Alan Schneider provided the algorithms necessary to create a celestial navigation calculator using the Hewlett Packard Model 65 programmable microcomputer [Ref. 8]. The peripheral device under design in this paper was intended to hit the mark somewhere between the HP-65 calculator and the submarine celestial navigation computer--much closer to the HP-65 side of the spectrum.

The HP-65 calculator solution to navigation problem lacks the same sort of smoothness that placing a square block into a round hole does; with a lot of effort it works but it leaves a lot to be desired. The HP-65 general purpose calculator uses programmable magnetic cards to hold prestored



programs that are executable on the machine. Though the device is relatively inexpensive the following drawbacks are noted:

1. Storage and maintenance of the program cards are cumbersome.
2. I/O is nonexistent.
3. Data format is not "conventional."
4. Record keeping must still be done manually.
5. The device is subject to pilferage.
6. Expansion of programs is difficult in its 100-instruction memory.

The advantages are the following:

1. Self contained (excluding the magnetic cards).
2. Easier to operate than most terminals.
3. Inexpensive.
4. Good supporting vendor.
5. Applicable to many other mathematical problems.



### III. DESIGN PHILOSOPHY

#### A. OBJECTIVES

The objectives of the design of the device are simple:

1. Implement Nuese and Schneider's algorithms in a device that does not have the disadvantages of the HP-65.
2. Maintain the advantages of the HP-65 without appreciably increasing the cost/performance factor.
3. Provide for integration into the overall information distribution system.
4. Provide the fail safe/soft features and hardcopy printout capabilities.

Two definitions will be stated at this time.

Microcomputer: A small general purpose computer. It contains a microprocessor and a small number of (support) LSI circuits. Usually mounted on one or two printed circuit boards. May or may not be programmable.

Minicomputer: A computer with larger word sizes and faster execution times than a microcomputer.

The proposal to specify a microcomputer for the celestial computer was based on the further observations that the microcomputer can be exactly specified as to the I/O structure, memory structure and contents, power supply designs, etc., whereas the minicomputer requires adjusting of all of the above mentioned properties to "fit" the minicomputer. The cost for a microcomputer is also much less than that of the minicomputer.

An eight-bit-wide microcomputer consisting of a microprocessor, a 128 x 8 bit RAM, a 1024 x 8 ROM and peripheral





adapter cost around \$473 in 1974, and by 1976 is estimated to cost under \$100 [Ref. 6]. Though larger than the HP-65 it is still less than ten times as large. An examination of data equipment suppliers will show that a small mini-computer will cost around \$5,000 and higher.

The device must be self-contained (except for power), easy to operate and relatively inexpensive. The I/O structure must allow for keyboard entry, and the formats for data must be naturalized ( $36^{\circ} 15' 33''$  vice  $36.1533^{\circ}$ ) to the problem. Note that the algorithm expressed in Appendix A uses the latter form due to limitations of the BASIC language I/O structure. As the device would be a small production lot, vendor support would be minimal or nonexistent; yet the simplicity of the device should allow for easy onboard maintenance.

## B. PHYSICAL DESIGN SPECIFICS

The device must be housed in one unit. The unit must allow for connection to power and to the data distribution system. The unit will have a simple ten-digit keyboard with a N/E (north or east) and S/W (south or west) key. A 16 element five-by-seven dot LED must be placed above the keyboard for verification of entries as the line printer prints a completed entry.

A 30 character wide line printer (for printing instruction, entries and calculation results) must be placed in the unit. The unit is illustrated in Figure 1, and a sample output strip is displayed in Appendix B.



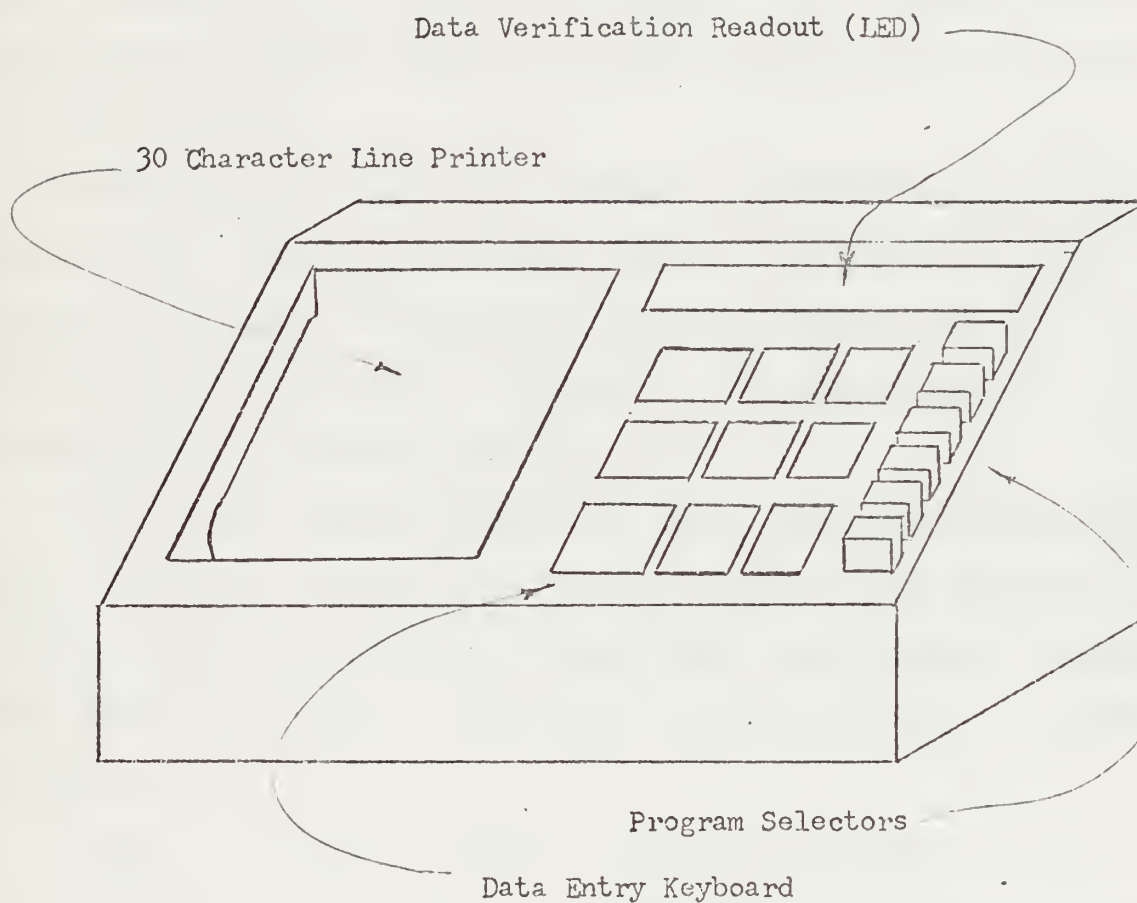


Figure 1-Celestial Navigation Microcomputer



#### IV. THE CELESTIAL FIX ALGORITHM

The purpose of the proposed navigational computer is to enhance the speed, accuracy and ease with which calculations such as those used in celestial navigation can be made. The program presented in Appendix A removes the need for the user to depend on the Nautical Almanac [Ref. 13] and other required H.O. publications by computing the values supplied by these tables.

When taking a fix several phases of sight reduction are considered: insertion of user values, correction of the sextant altitude, establishment of the position of the observed celestial body, solution of the navigational triangle and displaying the results.

User values include Greenwich date, day, month and year, the observer's assumed position, the name of the body observed, the time (Greenwich Mean Time), the sextant altitude, the observer's height of eye or dip and the sextant's index of correction.

Correction of the sextant's altitude includes changes in altitude due to refraction, semi-diameter of the sun and moon and parallax for the moon.

Establishment of the position of the observed celestial body on the earth (GP) involves the calculation of the Greenwich Hour Angle (GHA) and declination (Dec) of the body (see Figure 2). Associated with each star are eight



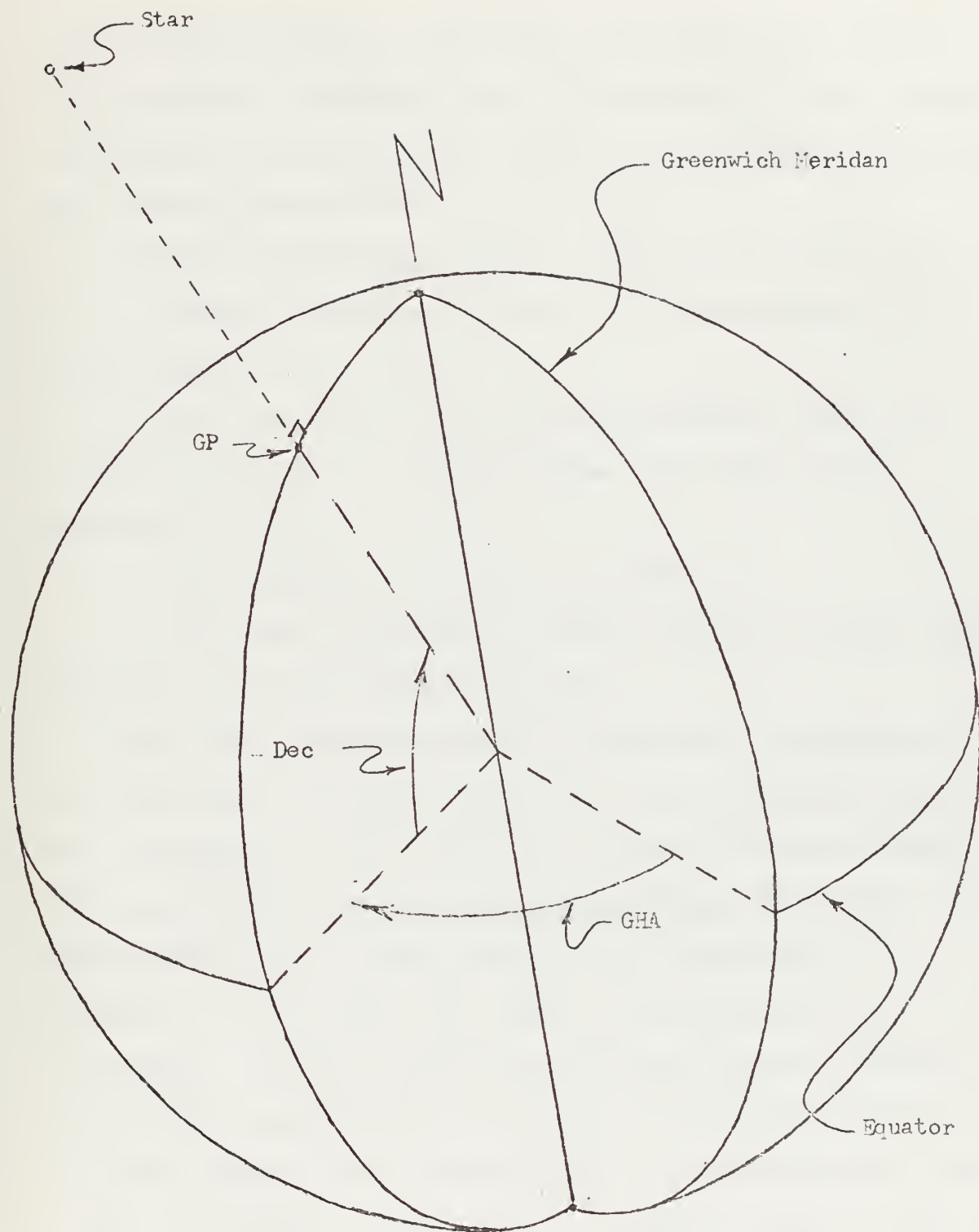


Figure 2-Geographic Position (GP) of a Star





intermediate constants which are calculated and combined with universal constants (rate of precession of the equinox, rotation of the earth on its axis and aberration) to produce the GHA and declination.

Constants which apply to all stars are the following:

1. Rate of rotation of the earth with respect to the vernal equinox [Ref. 11].
2. The longitude rate of the sun based on the time between successive crossings by the sun of the vernal equinox [Ref. 10].
3. The obliquity for epoch 1975 [Ref. 10].
4. The annual precession rates for epoch 1975 in right ascension and declination [Ref. 10].

With these common constants, particular intermediate constants associated with each star are developed. Utilizing the apparent right ascension and declination as found in Apparent Places of the Fundamental Stars [Ref. 14], right ascension and declination can be computed to a reference time through the effects of precession and aberration. Other constants include the rate of increase of the mean sun longitude in degrees per day, precession in right ascension and declination in degrees per day, true right ascension and declination in degrees, and the time behavior of the corrections for aberration [Ref. 8].

When the intermediate constants are developed and applied to the algorithms, the computed GHA and declination of a star will result. Discussion of the computation of



the constants, GHA and declination is presented in the next section. With the algorithm for computing GHA and declination of a star implemented it now becomes a matter of selecting those stars necessary to perform a sight reduction yielding a navigational fix.

Considering a three-star fix, the GHA and declination for each star is obtained as discussed above. Working with pairs of stars the solution of the associated navigational triangles will produce intersection points on the circles of equal altitudes (see Figure 3).

Three triangles are used to calculate a possible position as shown in Figure 4. The first triangle has vertices N (north pole), A and B. The angle A is either  $G_1 - G_2$  or  $360^\circ - (G_1 - G_2)$ . This angle represents the difference in the GHA's. Given the angle A and cosides  $d_1$  and  $d_2$  (the declinations), the law of cosines is applied to solve the third side  $t$ . Given the three cosides the angle B opposite coside  $d_1$  is calculated. Given three cosides in the second triangle,  $H_1$ ,  $H_2$  and  $t$ , the angle  $B_3$  opposite  $H_1$  is found.

Letting  $m$  be the sign of the  $\sin(G_1 - G_2)$ , the angles  $B(1) = mB - B_3$  and  $B(2) = mB + B_3$  are found. Given angle  $B(1)$  and cosides  $d_2$  and  $H_2$ , the third coside  $L(1)$  is found. Given the three cosides  $d_2$ ,  $H_2$  and  $L(1)$  the angle  $A(1)$  opposite  $H_2$  is found. In the same manner  $L(2)$  and  $A(2)$  are found.  $L(i)$  is equivalent to the latitude and  $A(i)$  is equivalent to the longitude of an intersection of the circles of equal altitude.



Circles of  
Equal  
Altitude

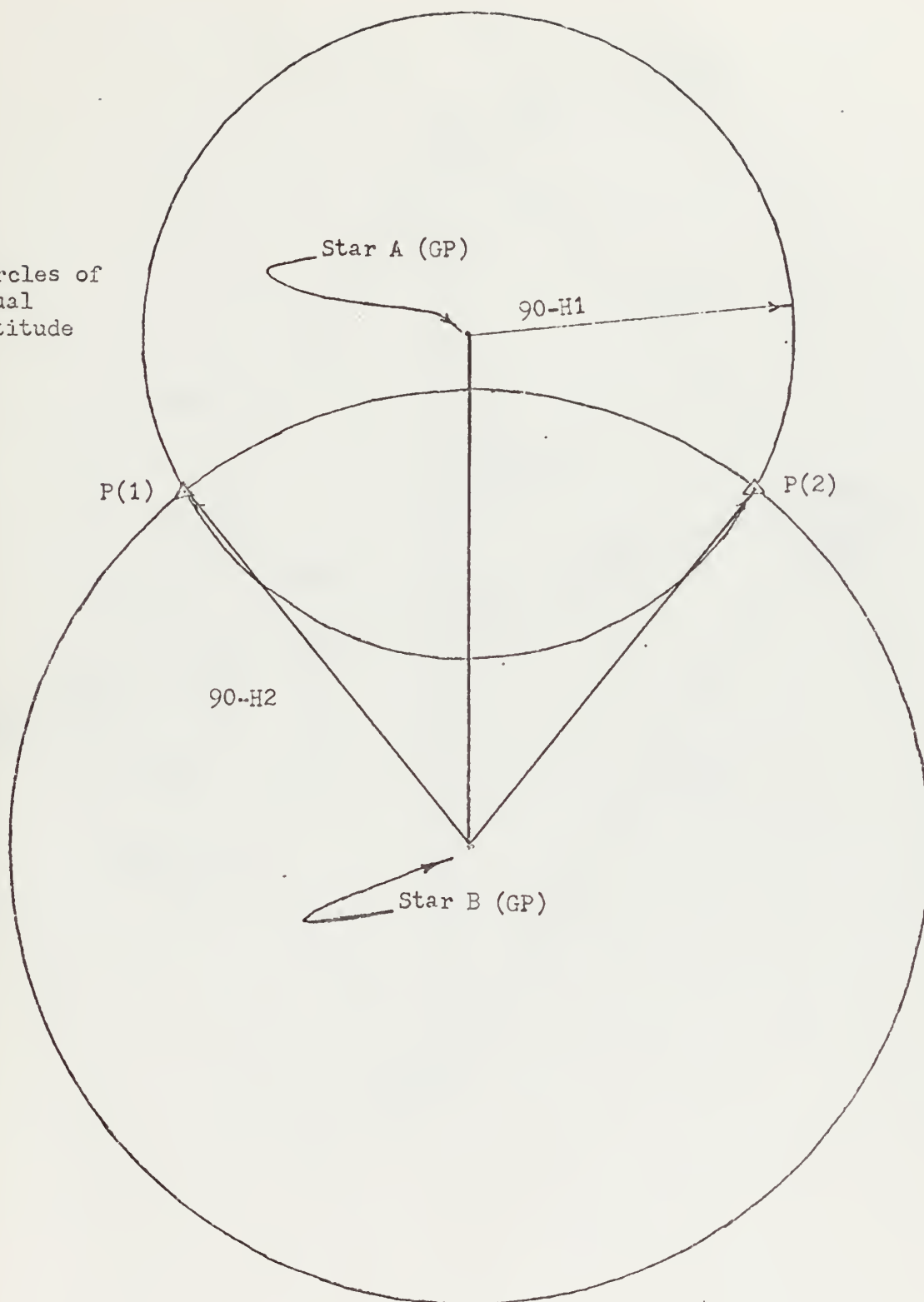


Figure 3--Navigational Fix from Altitudes  
of Two Celestial Bodies



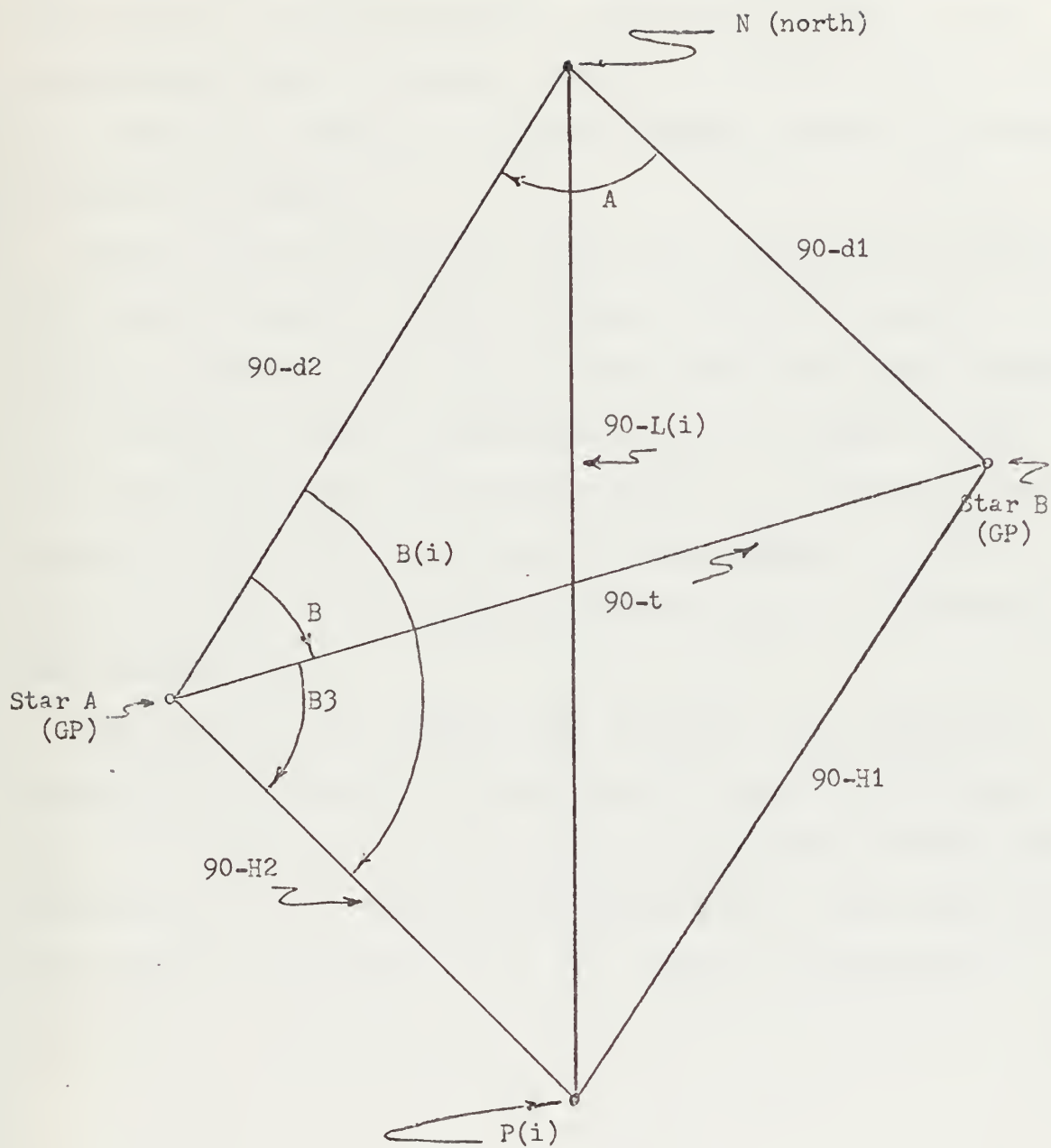


Figure 4-Triangles used for Navigational Fix





Repeating the algorithm considering the third celestial body and one of the previously used bodies, a second pair of positions will be found. A third repetition yields two more positions totaling six (the maximum) possible positions, as shown in Figure 5.

These six positions are then investigated to determine the closest three by determining the shortest distance connecting these three points. With this resulting triangle the location of the two endpoints of a line which is one-third away from any side of the triangle is computed. The center of this line containing the two endpoints is the center of mass of the triangle and therefore is considered the celestial fix.

The program will provide for accuracy in mathematical manipulation of numbers to one part in four billion using the BASIC interpreter at the Naval Postgraduate School and implementing the program on the INTEL 8080. This will provide for the accuracies of .1 or .2 miles as presented in Ref. 8.



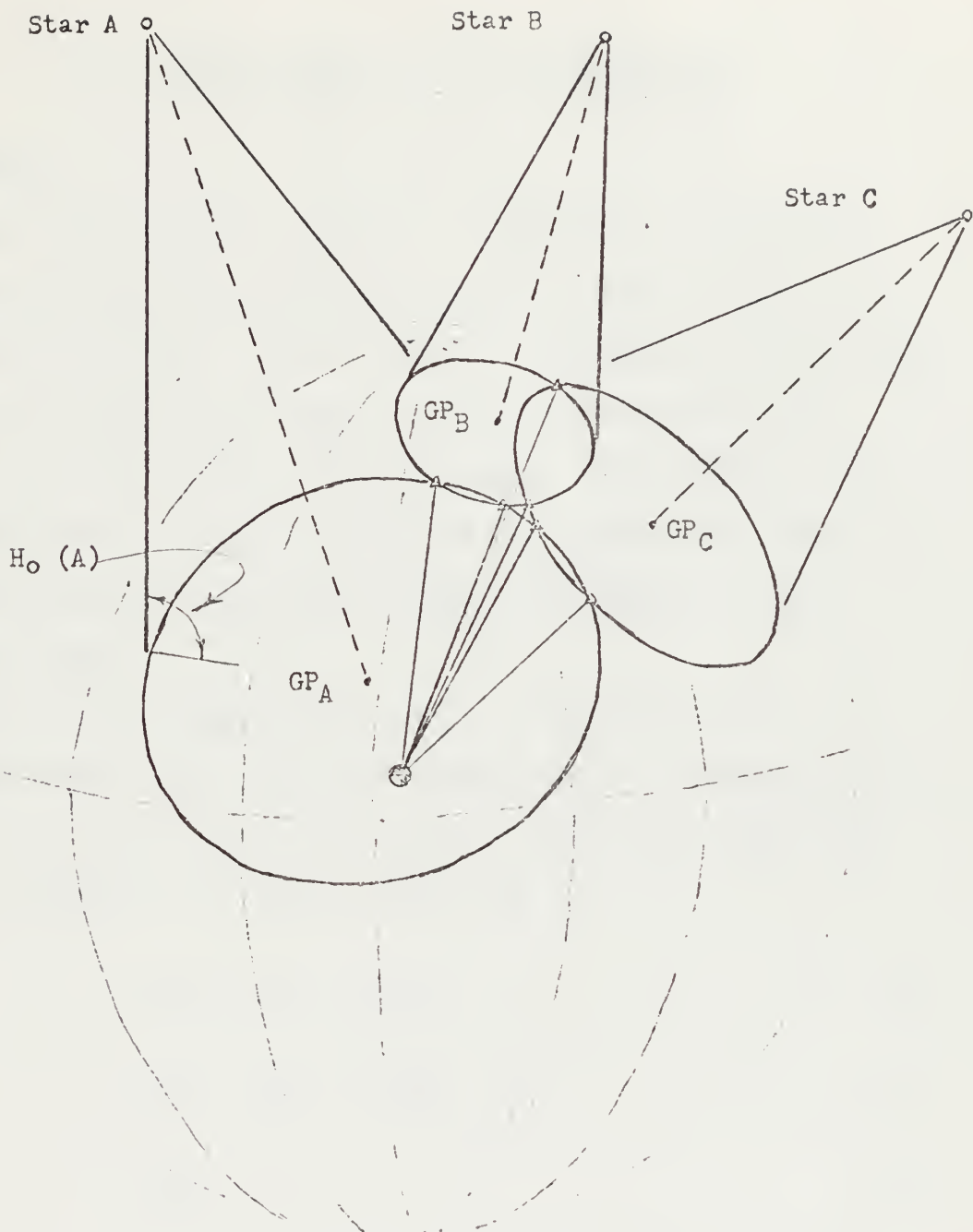


Figure 5-Overall Celestial Situation  
with six intersection points depicted



## V. PROGRAM THEORY AND IMPLEMENTATION

### A. THEORY

When computing the constants needed for the program a common reference time is necessary to provide for the accuracy of results. If the data is referenced to the mean equinox and equator of date, or to a fixed epoch then no error is introduced. The reference time employed in the algorithms is 0000 29 Nov 1971 as suggested by Ref. 8. Any data known at some other time is retarded to this reference time.

Note that the GHA of the star is related to the Greenwich Hour Angle of Aries ( $GHA_{\gamma}$ ) and the sidereal hour angle of the star (SHA), the latter being  $360^{\circ}$  minus the right ascension  $\alpha$ . Thus for any time  $t$ ,

$$GHA = GHA_{\gamma} + SHA \quad (1)$$

$$GHA = GHA_{\gamma} + (360 - \alpha) \quad (2)$$

$$GHA = GHA_{\gamma} - \alpha \quad (3)$$

The  $GHA_{\gamma}$  increases at a constant rate, namely, the rate of rotation,  $a$ , of the earth with respect to the vernal equinox. Knowing  $GHA_{\gamma}$  at some time  $t_1$ , then

$$GHA = a(t - t_1) + GHA_{\gamma 1} - \alpha(t) \quad (4)$$



where  $t$  is the present time measured in days and decimal parts of a day from the time  $t_1$ , a time in the recent past. Because it will later prove efficient to refer all events to an earlier time ( $t_0 = 0000\ 29\ \text{Nov}\ 1971$ ) this leads to

$$\text{GHA} = a(t - t_0) - a(t - t_1) + \text{GHA}_{\gamma 1} - \alpha(t) \quad (5)$$

$$= a(t - t_0) + \text{GHA}_{\gamma 0} - \alpha(t) \quad (6)$$

$$\text{where } \text{GHA}_{\gamma 0} = \text{GHA}_{\gamma 1} - a(t_1 - t_0) \quad (7)$$

The apparent right ascension  $\alpha$  at time  $t$  is related to its known true value  $\alpha'$  at some previous time  $t_1$  through the effect of precession and aberration.

Therefore (see "Supplement," [Ref. 11, pp. 39-49])

$$\alpha(t) = \alpha'(t_1) + \dot{\alpha}(t - t_1) + \Delta\alpha(t) \quad (8)$$

where

$\alpha'(t_1)$  = the true right ascension at time  $t_1$

$\dot{\alpha}$  = precession in right ascension (degrees per day)

$\Delta\alpha$  = correction in right ascension at time  $t$  for aberration (as defined on p. 47, "Supplement")

In order to refer all data to the common reference time  $t_0$ , it is convenient to rewrite (8) as follows:





$$\alpha(t) = \alpha'(t_1) + \dot{\alpha}(t - t_0) - \dot{\alpha}(t_1 - t_0) + \Delta\alpha(t) \quad (9)$$

$$= \alpha'(t_0) + \dot{\alpha}(t - t_0) + \Delta\alpha(t) \quad (10)$$

where

$$\alpha'(t_0) = \alpha'(t_1) - \dot{\alpha}(t_1 - t_0) \quad (11)$$

Inserting (10) into (6),

$$\text{GHA} = \text{GHA}_{\gamma_0} + a(t - t_0) - \alpha'(t_0) - \dot{\alpha}(t - t_0) - \Delta\alpha(t) \quad (12)$$

$$= [\text{GHA}_{\gamma_0} - \alpha'(t_0)] + (a - \dot{\alpha})(t - t_0) - \Delta\alpha(t) \quad (13)$$

Or,

$$\text{GHA} = a_{18} + a_{19}(t - t_0) - \Delta\alpha(t) \quad (14)$$

where

$$a_{18} = \text{GHA}_{\gamma_0} - \alpha'(t_0) \quad (15)$$

$$\begin{aligned} &= \text{GHA}_{\gamma_1} - a(t_1 - t_0) \\ &\quad - \alpha'(t_1) + \dot{\alpha}(t_1 - t_0) \end{aligned} \quad (16)$$

and

$$a_{19} = a - \dot{\alpha} \quad (17)$$

Now  $\Delta\alpha$  must be developed into a usable form. The equation on pages 47 and 48 in "Supplement" [Ref. 8] shows that aberration at time  $t$  is given by

$$= -k \sec \delta' [\cos \alpha' \cos \lambda \cos \epsilon + \sin \alpha' \sin \lambda] \quad (18)$$

where  $\lambda$  is the longitude of the mean sun measured along the



ecliptic from  $\gamma$ ,  $\epsilon$  is the obliquity (angle between the mean planes of the ecliptic and equator),  $k$  is the constant of aberration defined on page 48 in "Supplement," having a value of  $20.47''$ , and  $\alpha'$  and  $\delta'$  are the true right ascension and declination, all values being at time  $t$ . Selecting  $t_1$  as a convenient time for determining the mean place of the star and the obliquity, and regarding  $\alpha'$ ,  $\delta'$  and  $\epsilon$  as constants over the interval from  $t_1$  to  $t$ , equation (18) becomes:

$$\begin{aligned} \Delta\alpha = & - \left( \frac{20.47}{3600} \sec \delta'_1 \cos \alpha'_1 \cos \epsilon_1 \right) \cos \lambda \\ & - \left( \frac{20.47}{3600} \sec \delta'_1 \sin \alpha'_1 \right) \sin \lambda \end{aligned} \quad (19)$$

$$= a_{10} \cos \lambda + a_{11} \sin \lambda \quad (20)$$

where

$$a_{10} = - \frac{20.47}{3600} \sec \delta'_1 \cos \alpha'_1 \cos \epsilon_1 \quad (21)$$

$$a_{11} = - \frac{20.47}{3600} \sec \delta'_1 \sin \alpha'_1 \quad (22)$$

The factor 3600 enters because  $\Delta\alpha$  will be computed as fractions of a degree whereas  $k$  was given in arc seconds.

Equation (20) can be rewritten as

$$\Delta\alpha = a_{12} \sin (\lambda + a_{13}) \quad (23)$$



where

$$a_{12} = \sqrt{a_{10}^2 + a_{11}^2} \quad (24)$$

and

$$a_{13} = \tan^{-1} \left( \frac{a_{10}}{a_{11}} \right) \quad (25)$$

The longitude of the mean sun increases linearly with time according to the equation

$$\lambda(t) = \dot{\lambda}(t - t_1) + \lambda_1 \quad (26)$$

where  $\lambda_1$  is the known longitude at some convenient reference time  $t_1$ . As has previously been done, it is best to refer to a common reference time  $t_0$  by rewriting (26) as follows:

$$\lambda(t) = \dot{\lambda}(t - t_0) - \dot{\lambda}(t_1 - t_0) + \lambda_1 \quad (27)$$

$$= \dot{\lambda}(t - t_0) + \lambda_0 \quad (28)$$

where  $\lambda_0$  is defined as follows:

$$\lambda_0 = \lambda_1 - \dot{\lambda}(t_1 - t_0) \quad (29)$$

Substituting (28) into (23), the results are

$$\Delta\alpha = a_{12} \sin [\dot{\lambda}(t - t_0) + \lambda_0 + a_{13}] \quad (30)$$

$$= a_{12} \sin [\dot{\lambda}(t - t_0) + a_{14}] \quad (31)$$



where  $a_{14}$  is defined as follows:

$$a_{14} = \lambda_0 + a_{13} \quad (32)$$

Using (31) in (14) the substitution yields

$$\text{GHA} = a_{18} + a_{19}(t - t_0) - a_{12} \sin [\dot{\lambda}(-t_0) + a_{14}] \quad (33)$$

Because  $a_{19}$  includes earth rate, roughly  $360^\circ$  per day, and that  $t - t_0$  may be on the order of 30,000 days, to preserve accuracy in the computation it is worthwhile to separate  $a_{19}$  into two parts, one of which,  $a_{21}$ , is  $360^\circ$  and the other  $a_{22}$ , is the remainder. Let

$$a_{19} = a_{21} + a_{22} \quad (34)$$

where

$$a_{21} = 360 \quad (35)$$

and

$$a_{22} = a_{19} - 360. \quad (36)$$

In similar fashion, denote the number of days from the reference  $t_0$  to the time  $t$  to be  $D.d$  where  $D$  is a whole number of days since 0000 Nov 29, 1971, and  $d$  a fractional part. That is

$$t - t_0 = D.d \quad (37)$$

Then it is obvious that the product of the angular rate times the time yields an angle modulo  $360^\circ$  which can be expressed





as follows:

$$a_{19}(t - t_0) = (360 + a_{22})(D + d) \quad (38)$$

$$= 360D + a_{22}(D + d) + 360d \quad (39)$$

$$= a_{22}(D + d) + 360d \quad (40)$$

$$= a_{22}(D.d) + 360d \quad (41)$$

Substituting this result and (37) into (33) yields

$$\text{GHA} = a_{18} + a_{22}(D.d) + 360d - a_{12}[\dot{\lambda}(D.d) + a_{14}] \quad (42)$$

The constants  $\dot{\alpha}$  and  $\dot{\delta}$ , precession in right ascension and declination, for a particular star are obtained using the formulas on page 39 in "Supplement," except that the values of the constants have been taken for epoch 1975 from the American Ephemeris and Nautical Almanac [Ref. 10]. Conversion to degrees per day is

$$\dot{\alpha} = \frac{1}{3600 \times 365.24219418} [46.1060 + 20.0404 \sin \alpha'_1 \tan \delta'_1] \quad (43)$$

$$\dot{\delta} = \frac{1}{3600 \times 365.24219418} [20.0404 \cos \alpha'_1] \quad (44)$$

This now completes the derivation for the Greenwich Hour Angle of a star at time  $t$ .

Turning attention to the derivation of declination, it is noted that the derivation is similar to those just carried out. Declination of a star changes because of precession and parallax. Letting  $\delta$  be the apparent



declination at time  $t$ ,  $\delta'$  be the true declination at some specified time,  $\dot{\delta}$  be the rate of precession of this star in declination,  $\Delta\delta$  be the correction to declination for aberration, the result is

$$\delta(t) = \delta'(t_1) + \dot{\delta}(t - t_1) + \Delta\delta(t) \quad (45)$$

where  $t_1$  is the reference time of which the true right ascension and declination are known. The numerical value of  $\dot{\delta}$  is given in (44). This form is converted to the standard reference time  $t_0$  by the same method which has been used above. The result is

$$\delta(t) = \delta'(t_0) + \dot{\delta}(t - t_0) + \Delta\delta(t) \quad (46)$$

where

$$\delta'(t) = \delta'(t_1) - \dot{\delta}(t_1 - t_0) \quad (47)$$

From pages 47 and 48 in "Supplement" the time behavior of the correction for aberration is

$$\begin{aligned} \Delta\delta &= -k(\sin \epsilon_1 \cos \delta'_1 - \cos \epsilon_1 \sin \alpha'_1 \sin \delta'_1) \cos \lambda \\ &\quad - k \cos \alpha'_1 \sin \delta'_1 \sin \lambda \end{aligned} \quad (48)$$

$$= a_{15} \cos \lambda + a_{16} \sin \lambda \quad (49)$$

where

$$a_{15} = \frac{20.47}{3600} (\sin \epsilon'_1 \cos \delta'_1 - \cos \alpha'_1 \sin \alpha'_1 \sin \delta'_1) \quad (50)$$

$$a_{16} = \frac{20.47}{3600} \cos \alpha'_1 \sin \delta'_1 \quad (51)$$



The constants  $a_{15}$  and  $a_{16}$  are in degrees. Simplifying further

$$\Delta\delta = a_{17} \sin \left( \lambda + \tan^{-1} \frac{a_{15}}{a_{16}} \right) \quad (52)$$

where

$$a_{17} = \sqrt{a_{16}^2 + a_{15}^2} \quad (53)$$

Substituting (52) and (28) into (46),

$$\delta(t) = \delta'(t_0) + \dot{\delta}(D.d) + a_{17} \sin[\dot{\lambda}(t - t_0) + a_{20}] \quad (54)$$

where

$$a_{20} = \lambda_0 + \tan^{-1} \frac{a_{15}}{a_{16}} \quad (55)$$

and

$$t - t_0 = D.d \quad (56)$$

The final equations of importance which lead to GHA and declination are (42) and (54). Note that eight star-particular constants are involved in these two equations, plus  $\dot{\lambda}$ , a constant which appears in all calculations of GHA and declination of the stars.

The equations involving reference times have been set up in such a way as to maximize freedom in their selection. Reference time  $t_0$  was selected to be 0000 Nov 29, 1971, to facilitate the calculation of time in days with a minimum number of entries as suggested by Ref. 8.



The reference time  $t_1$  is any convenient time, presumably in the recent past, at which the GHA of Aries is precisely given in some reference tabulation. The time  $t_1$  is referenced to true right ascension and declination as listed in some reference almanac. The true right ascension and declination at  $\alpha'(t_1)$  and  $\delta'(t_1)$  can be obtained from the apparent values  $\alpha(t_1)$  and  $\delta(t_1)$  by setting  $t = t_1$  in equations (8) and (45) and solving for  $\alpha'$  and  $\delta'$ .

The reference time  $t_1$  is also the time, again in the recent past, at which obliquity  $\epsilon_1$ ,  $\alpha'_1$ ,  $\delta'_1$  and  $\lambda_1$  are accurately known. The reference time for  $\lambda_1$  is such that the accuracy of the sun's mean longitude is known from some almanac data. It will minimize the errors due to (a) eccentricity of the earth's orbit, and (b) the leap year cycle, if  $t_1$  is chosen at or near the time of earth perihelion, in the midpoint of a leap year cycle.

It should be noted that  $t_1$  for the known value of GHA of Aries may be different from the  $t_1$  for the known values of true right ascension and declination. Likewise  $t_1$  may be yet another time for the known sun's mean longitude. Since all the values (constants) associated with  $t_1$  are retarded to  $t_0$  no difference in  $t_1$  is objectionable.

## B. IMPLEMENTATION

The program presented in Appendix A is the result of several conversions. The first conversion consisted of taking algorithms expressed in Ref. 8 with the HP-65





programs to produce a debugged FORTRAN IV program. The HP-65 programs deal with degrees as a unit of angular measure vice radians. A conversion of all data from degrees to radians was made for use in the FORTRAN IV and BASIC programs. General routines 0010 and 0020 provide for I/O conversion of radians to degrees, and hours, minutes, and seconds to hours (and vice versa). These conversions were needed for ease of data entry and data output.

The HP-65 programs are compact due to the limited number of registers used for intermediate storage. This efficiency of memory was carried through to the FORTRAN IV programs and then to the BASIC programs for the routines that determine the GHA and declination of the star and the subroutine that determines the azimuth ( $Z_n$ ) and distance (subroutine 0300 and 0400, respectively). It was felt that this level of efficiency need not be continued in the remainder of the program due to the need for self-documentation.

FORTTRAN IV was selected as the intermediate step in the conversion due to the amount of core memory allocated to the FORTRAN compiler. The BASIC interpreter is limited in memory on the IBM 360 installation and would not hold the completed BASIC program.

The reduced precision of the FORTRAN IV version of the program (as it was not programmed in double precision) was attributable to the internal formatting of the variables. High precision was not of great concern at this stage of development.



As the goal of this process was to provide a program that would implement the algorithms necessary to complete a celestial fix on a microprocessor, the FORTRAN IV version was converted to BASIC. The BASIC program allows for much greater precision in computation as the values are expressed in 32 bit floating point, when interpreted by the available BASIC interpreter in the Microcomputer Laboratory at the Naval Postgraduate School. The BASIC interpreter at the W. C. Church Center was not of sufficient size to run the entire program. Therefore, the main subroutines and all required support routines were run and debugged separately, and required information was transferred manually from routine to routine. Because the BASIC language does not provide for scopal variables and due to the limited types of identifiers available much of the self-documentation of the FORTRAN IV program was lost. Table I correlates variables in subroutine 0100 in the program listing (Appendix A) to those variables previously discussed during the theory presentation.

Subroutine 0100 computes the eight constants associated with a star and places them in array C (elements C(2)-C(9)). Element C(1) is the rate of increase of the mean sun longitude (common to all stars).

In order to organize the variables as much as possible a scheme was invoked as explained below. Since there were a number of subroutines in the program, a method of passing variables was constructed. The variables R1-R9 and P1-P9



TABLE I

Identifier Correlation of Variables in Subroutine  
0100 and Equation Variables

Subroutine Variable	Equation Variable
C(1)	$\dot{\lambda}$
C(2)	$a_{20}$
C(3)	$a_{17}$
C(4)	$\dot{\delta}$
C(5)	$\delta'(t_0)$
C(6)	$a_{22}$
C(7)	$a_{12}$
C(8)	$a_{18}$
C(9)	$a_{14}$
I3	$\alpha_1$
I4	$\delta_1$
I5	$\dot{\alpha}$
I6	$a_{19}$
I7	$a_{10}$
I9	$a_{11}$
J0	$a_{15}$
J1	$a_{16}$
C5	$\epsilon_1$



are passing variables. Variables used in three or more non-general routines were considered as global and were selected to be as self-documenting as possible. For example, the two constants for each star are placed in the array  $S(X,X)$ . The constants used by routine 0300 to determine GHA and declination which are created from the S array are placed in the C vector. All other constants were represented by the format C1-C9.

Several routines use intermediate variables while calculating the object variables. These variables are represented by I1-I9, J1-J9 and M1-M9. All logical values are represented by L1-L9. Note that this scheme was invoked to eliminate problems created by not having scopal variables. This must be considered in the selection of a language for other peripherals used in the system. Points to be considered in selecting a final programming language for such a system and the peripherals attached to it are discussed in Section VII.





## VI. THE INFORMATION DISTRIBUTION SYSTEM

### A. OBJECTIVES

This subject will primarily be concerned with the integration of a general purpose computer into the existing shipboard information flow. The "system" in the spotlight is the navigation suite aboard U.S. Navy ships. The philosophy for this approach is simple and is best stated by Joseph Weir [Ref. 1]:

"Good systems can be designed only in the presence of potential users and applications. By using the system in numerous applications and assiduously watching its performance, it gradually can be adopted to the (actual) needs of the application and the user. Ingenious ivory tower solutions frequently do not work or they solve problems that don't exist."

A close examination of this statement while relating it to the navigation suites will show that in fact the "system" has been designed both as far as it need and can be with the existing equipment. Note that the "system" comprises geographically dispersed equipment and telephones for communication links.

The application and needs of the navigation system have been defined as will be discussed later in this section. If all this is true then why rock the boat by imposing a system, apparently complex, on an existing "system"?

Discussion with crewmen aboard the USS Hancock indicated that the existing system is cumbersome, subject to human frailties and is not timely enough for the devices



on board. OMEGA radio navigation sets are capable of providing information for navigational fixes every few seconds, yet this information is only used hourly. In this case the communication link is the "weak" link, and to eliminate this weak link an information distribution system must be imposed. The distribution system must appear translucent and straightforward to the user, for as Weir mentions,

"The more complex the system, the more likely it is that it won't get used or will be misunderstood or misused." [Ref. 1]

All systems have sensors and effectors, and some communication constraints placed on them. Although most of these sensors for the navigation system in fact exist, there are always new devices being created.

For example, ten years ago operational OMEGA navigation sets did not exist. If a system had been created and implemented on the ships at that time without forethought with regard to new devices, then the information system would not have been able to integrate the new device into the network.

In order to fully utilize an information distribution system there must be "terminals" with which to input information and query the system. For example, suppose the bridge wants to know the average RPM of the four propulsion shafts. Engineering would have a terminal (either automated or manual) that would accept certain required data, e.g., tachometer inputs, transform it into its varied forms and relay the



data to the central processor. The processor would then transfer the required information to the terminal at the bridge.

## B. DESIGN PHILOSOPHY

The question might be asked, "Why try to integrate an information distribution system with the existing peripherals?" The answer should be obvious--cost of implementation. Existing equipment must be fully exploited until it is not wise to continue using it.

By integrating existing equipment into a network instead of creating all of the component parts of the system anew, the cost of research and development of all the new sensors and effectors in the system can be elided [Ref. 2]. Of course, the use of current equipment will demand the design of interfacing equipment such as analog to digital converters, digital to analog converters, fault sensing networks, etc. However, the cost of these units should be relatively inexpensive when compared to the redesign of any one sensor or effector in the system. The application of the same item (A/D or D/A, etc.) to many peripherals is a cost savings when design efforts are considered.

As distribution of existing data is the concern of the central system and since there is a multiplicity of like data, the question of which source's information should be distributed must be handled. A rather trite answer is "the best." Again a question, "Which sensor is providing 'the best' information?"



A definition of the term "best" is recursive and must include words like timeliness, accuracy and statistical validity. As each sensor in the system is operating at pre-system design rates, some periodic (auto track LORAN C) and others at non-periodic rates (celestial sightings), timeliness takes on a meaning associated with the quality of the source. For example, if the only peripheral capable of input to the system is the celestial navigation unit, an output of ship's position would only be considered timely if the last celestial position had been updated with inputs from the "best" heading and speed information to provide an estimate of the current position.

The word "accuracy" when associated with "best" means most accurate on a timely basis. A position based on a ten-hour old OMEGA position is usually not as accurate as a five-minute old celestial position.

This is all well and good, yet how are these descriptors compiled? A central machine is obviously the device that will have to provide the software to meet problems based on statistical properties.

Statistical properties of each measuring device must be taken into account. The proper error probabilities must be applied to each input; then based on the result of the combined statistics, the "best" source is selected to represent that class of input. It is this source's information (corrected for time) that must be distributed.





Along with statistical constraints imposed on the data, a status check must also be imposed on the device. If the data is apparently correct yet the device has failed, then the data is obviously of no use. The status determination system must be as nearly fail-safe as possible. Since this paper is proposing a centralized star network vice a fully connected or ring network [Ref. 2], all the status checks must be initiated at the central computer (see Figure 6). In order to design a fail-safe status monitor, the following considerations must be employed:

1. Notice of failure of the central computer must be distributed to all peripheral stations.

2. Failure of each peripheral must affect only that peripheral and, at worst, place the system in a degraded mode.

3. Status of selected peripherals might have to be recorded for legal records.

4. Failure of the peripheral must not cause failure of the status check mechanism.

There are several reasons for designing an information distribution system. Many operations aboard a ship are dependent on the same source of information. For example, the Combat Information Center and radar stations need true course and speed information. These are parallel operations and their information must be based on the same source. A distribution system would be the answer to this problem.



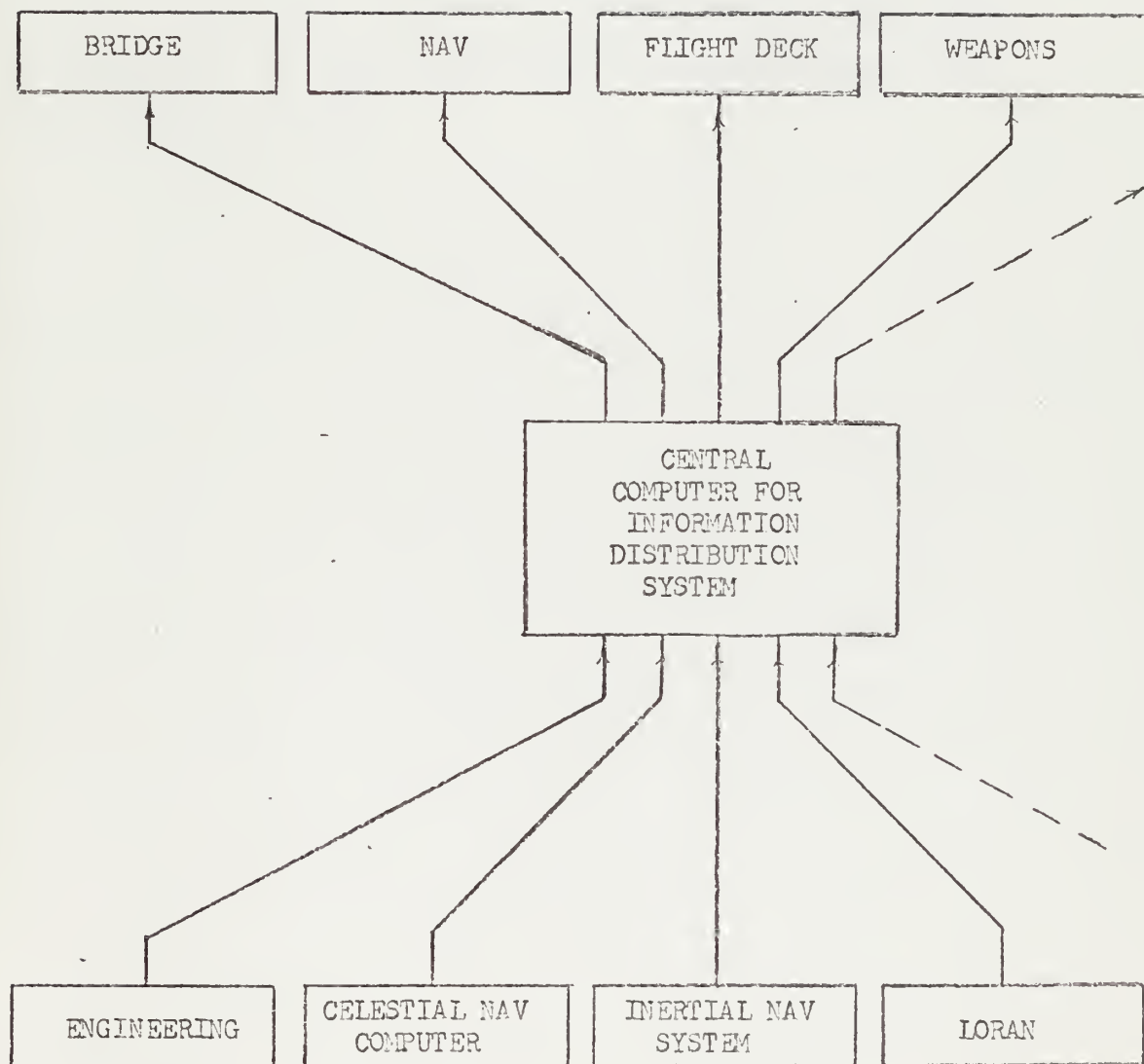


Figure 6-Information Distribution System



All of the discussion thus far has been about distribution of information gathered from sensors, so where is all this fine information used? An example may best describe the answer. Suppose the navigator updates the current ship's position with the celestial navigation computer. If the Captain desires to proceed to a desired point, the point could be placed into the system and the system would provide proper steering information (great circle steering) to the Helmsman, or this data could be transmitted to an auto-pilot upon request.



## VII. LANGUAGE SELECTION

The subject of language selection is placed at this point in the paper due to its applicability to both the celestial navigation microcomputer and an information distribution system as described within. The considerations in selecting a programming language for these two devices have much in common, the most important of which is the microprocessor.

There are two main considerations in selecting a programming language for a microcomputer. The first and most important consideration is that of efficiency; efficiency of both memory utilization and program execution speed. The second point concerns the cosmetics of the language (i.e., is the language easy to read and is it self-documenting?).

In consideration of the first property, that of efficiency, the language must be chosen in the light of cost/availability of volatile memory. Even though the addressing schemes of some microprocessors provide for large memories there may be physical and/or cost limitations placed on the microcomputer design that make very efficient use of the available memory imperative. For peripherals operating with unmodifiable code the number of memory locations can be optimized and placed with a high degree of efficiency. For processors using programs which have subroutines with arrays of data created during the





execution of the program, the placing of these arrays on a metal mask chip or fixing and reserving the addresses of these arrays is an inefficient use of memory.

The best way to program in an efficient manner is to use block structured programming techniques utilizing "scopal" variables. Block structured languages like Algol 60 and PLM [Ref. 12] allow for scopal variables to include arrays which provide for efficient use of random access memory and reduce the amount of programmable/read only memory. By creating arrays as needed, common random access memory may be used very efficiently allowing for lower investment in space and cost.

The second area of concern, that of cosmetics, is a growing concern with all languages. It may be a seemingly unimportant aspect until the prospects of non-volatile runtime modifiable LSI memories (e.g., ferroelectric memories [Ref. 15]), multiplicity of similarly constructed peripherals and common maintenance support are considered. The celestial navigation microcomputer program bases all of the calculations for geographical positioning of the body constants which vary from year to year. In order for the device to enjoy extended longevity, the constants must be alterable. The language must allow for the documentation to help provide this support. An assembler program is not a self-documenting program no matter how efficient it is. Its logic and



meaning are often forgotten with the loss of the programmer. Indeed, the programmer may lose the logic to his own program if he has been away from it for a length of time. For example, the statement below is obvious in its intent:

$$\text{NO\$YEARS} = \text{NO\$DAYS} / 365.$$

Yet even though the next statement provides for the same efficiency of calculation it does not connote as clear a meaning:

```
LIT DAYS; LIT 365; DIV; LIT YRS; STO; .....
```

The assembler language form of this statement loses the easily conceived meaning of the first statement.

Though the celestial fix program in Appendix A was written in the BASIC language due to the precision attainable through its use, the program itself would not be understandable without the comment statements and extra documentation.

Specifically, there are primarily two high level languages available to the microprocessor programmer; BASIC as implemented at the Naval Postgraduate School and PLM. BASIC, as the name implies, provides for the basic needs of a programmer. BASIC has subroutines, iterative and logical capability, the full range of mathematical manipulators and limited I/O. A quick glance at the program in Appendix A provides the evidence of a major drawback: lack of variable scope and identifiers. There are no local



variables; all variables are global. This property coupled with the limited identifier schemes causes a programmer of a large program to run out of variables quickly. His attention is turned to efficient use of identifiers instead of efficient programming techniques.

PLM, on the other hand, provides for scopal variables and identifiers that allow for self-documentation. The following statements are equivalent; however, the second connotes more meaning:

$$\text{LET J1} = \text{G(Q)} - \text{G(R)}$$
$$\text{DIFF} = \text{GHA1} - \text{GHA2}.$$

The array used in the determination of the three intersections used to calculate the fix (subroutine 0600 ... `ARRAY V(6,6)`) is a temporary array. In BASIC the array is permanently allocated space. Using PLM and scopal variables the array would be created by the program during execution (indeed, could be of variable size) and be destroyed or written over upon exit from the subroutine. There are a few drawbacks with the use of PLM. The primary drawback in the execution of the program in Appendix A is the word size. Sixteen bits do not provide the accuracy needed for the celestial navigation problem. Though this problem can be solved, it is not easily solved.

Special mathematical routines like trigonometric functions are not available in PLM, yet they can be programmed. The point is that PLM can be tailored to the needs of the



programmer with a little extra work while the BASIC language cannot be tailored to the user at all. PLM provides a much more versatile language yet the detail required in programming is greater than that of BASIC.

A final comment on the selection of a language deals with the multiplicity of processors on an information distribution system. If each processor were programmed at a different level with a different language the maintenance and design problems would be horrible. For example, suppose one processor is to provide a second processor information. If the first processor is programmed in BASIC then it would probably transmit its data in 32 bit floating point form. The second processor (programmed in PLM) would have to have an I/O routine to convert that processed data to a local format. If there were a third language then the PLM-programmed processor may need yet another I/O processing routine. This situation is the grounds for selection of a single programming language for a system of microprocessors.





## VIII. CONCLUSIONS

Currently, the proposed computer is capable of developing a navigational fix (without use of the Nautical Almanac and H.O. Publications) utilizing only the 57 navigational stars. Because of this limitation it is obvious that numerous and known applications are yet to be incorporated. Among these include methods which employ the attributes of the sun, moon and planets so that positioning with these bodies can be performed. Considering functional aspects for future development of programs, suggested programs are closest point of approach (CPA), great circle calculations, star location utilizing estimated altitude and azimuth and fix triangulation routines.

The sun, moon and planets present special problems because of their proximity to earth. They do not appear fixed in the sky as do stars. Stars do move across the sky due to the earth's rotation but their great distances impart a constant relationship to their neighboring stars. The planets and moon, being closer to earth, possess no fixed relation in the evening sky with the stars.

The size of the sun and moon requires a special consideration termed semi-diameter. More accurate sight reductions can be performed by aiming the sextant at the upper and lower limb of these bodies; corrections for the radius of the body can be programmed.



For lunar observations horizontal parallax is a parameter required to be considered. It is the apparent difference in the position of a body when viewed from two different locations. Horizontal parallax is significant for the moon because it is the closest body to earth.

The celestial lines of position are included to give the navigator the option of plotting LOP's. This method should provide the navigator with an azimuth, a distance and a toward/away indication in reference to the assumed position of the observer with which the LOP should be plotted. This is the backup routine to the fix calculation algorithm.

The proposed celestial navigation computer was used as a vehicle for the approach to a design of a dedicated micro-computer. Other devices of similar construction used as peripherals in the information distribution system would be approached in the same manner. Approaching the design of each device on its own merits will allow for the system to grow in a manner that "fits" the situation, i.e., a single "ivory tower" system would not result.

Indeed, the development of the central processor (assuming enough ports are planned for) would grow with the system as each new unit would simply add length to the polling routines and interrupt structures. Following along with Weir's natural growth concept, each processor's programs would be designed and built (or at least the algorithms be provided for) as needed by cognizant individuals, from



current literature, as done in the case of the celestial navigation computer. The only restraint placed on the design/designer would be that of language and interface specifications. As delineated earlier, the authors believe that of the two languages examined PLM would be the most efficient language and would provide for much needed self-documentation.

The central processor in the information distribution system may very well be a minicomputer or a microcomputer depending on its functions. It need only be a distribution device designed along the same lines as the peripherals it supports. The level of computation provided by the central processor must be established early in the design phase because the peripherals would need to "take up the slack" for the computational capabilities not provided by the central processor. For example, the averaging of the revolutions per minute may be accomplished at the peripheral (which is recommended) or at the central processor on the request of the inquirer. Again, keeping this device simple (only the necessary switching and monitoring routines) will provide for more independence and faster information transfer.



# CELESTIAL FIX PROGRAM LISTING

```
B(2),L(6),E(5),V(6,6),D(3),H(3),G(3)
C(9),Z(3),A(3),Q(3),S(2,57)
REM DEFINE LOGICAL VALUES AND CONSTANTS
T=1
F=0
C1=5
C2=57
C3=3
C4=EM
R(1)=LOAD
R(1,1)=0
R(1,2)=0
R(1,3)=0
R(1,4)=0
R(1,5)=0
R(1,6)=0
R(1,7)=0
R(1,8)=0
R(1,9)=0
R(1,10)=0
R(1,11)=0
R(1,12)=0
R(1,13)=0
R(1,14)=0
R(1,15)=0
R(1,16)=0
R(1,17)=0
R(1,18)=0
R(1,19)=0
R(1,20)=0
R(1,21)=0
R(1,22)=0
R(1,23)=0
R(1,24)=0
R(1,25)=0
R(1,26)=0
R(1,27)=0
R(1,28)=0
R(1,29)=0
R(1,30)=0
R(1,31)=0
R(1,32)=0
R(1,33)=0
R(1,34)=0
R(1,35)=0
R(1,36)=0
R(1,37)=0
R(1,38)=0
R(1,39)=0
R(1,40)=0
R(1,41)=0
R(1,42)=0
R(1,43)=0
R(1,44)=0
R(1,45)=0
R(1,46)=0
R(1,47)=0
R(1,48)=0
R(1,49)=0
R(1,50)=0
R(1,51)=0
R(1,52)=0
R(1,53)=0
R(1,54)=0
R(1,55)=0
R(1,56)=0
R(1,57)=0
R(1,58)=0
R(1,59)=0
R(1,60)=0
R(1,61)=0
R(1,62)=0
R(1,63)=0
R(1,64)=0
R(1,65)=0
R(1,66)=0
R(1,67)=0
R(1,68)=0
R(1,69)=0
R(1,70)=0
R(1,71)=0
R(1,72)=0
R(1,73)=0
R(1,74)=0
R(1,75)=0
R(1,76)=0
R(1,77)=0
R(1,78)=0
R(1,79)=0
R(1,80)=0
R(1,81)=0
R(1,82)=0
R(1,83)=0
R(1,84)=0
R(1,85)=0
R(1,86)=0
R(1,87)=0
R(1,88)=0
R(1,89)=0
R(1,90)=0
R(1,91)=0
R(1,92)=0
R(1,93)=0
R(1,94)=0
R(1,95)=0
R(1,96)=0
R(1,97)=0
R(1,98)=0
R(1,99)=0
R(1,100)=0
R(1,101)=0
R(1,102)=0
R(1,103)=0
R(1,104)=0
R(1,105)=0
R(1,106)=0
R(1,107)=0
R(1,108)=0
R(1,109)=0
R(1,110)=0
R(1,111)=0
R(1,112)=0
R(1,113)=0
R(1,114)=0
R(1,115)=0
R(1,116)=0
R(1,117)=0
R(1,118)=0
R(1,119)=0
R(1,120)=0
R(1,121)=0
R(1,122)=0
R(1,123)=0
R(1,124)=0
R(1,125)=0
R(1,126)=0
R(1,127)=0
R(1,128)=0
R(1,129)=0
R(1,130)=0
R(1,131)=0
R(1,132)=0
R(1,133)=0
R(1,134)=0
R(1,135)=0
R(1,136)=0
R(1,137)=0
R(1,138)=0
R(1,139)=0
R(1,140)=0
R(1,141)=0
R(1,142)=0
R(1,143)=0
R(1,144)=0
R(1,145)=0
R(1,146)=0
R(1,147)=0
R(1,148)=0
R(1,149)=0
R(1,150)=0
R(1,151)=0
R(1,152)=0
R(1,153)=0
R(1,154)=0
R(1,155)=0
R(1,156)=0
R(1,157)=0
R(1,158)=0
R(1,159)=0
R(1,160)=0
R(1,161)=0
R(1,162)=0
R(1,163)=0
R(1,164)=0
R(1,165)=0
R(1,166)=0
R(1,167)=0
R(1,168)=0
R(1,169)=0
R(1,170)=0
R(1,171)=0
R(1,172)=0
R(1,173)=0
R(1,174)=0
R(1,175)=0
R(1,176)=0
R(1,177)=0
R(1,178)=0
R(1,179)=0
R(1,180)=0
R(1,181)=0
R(1,182)=0
R(1,183)=0
R(1,184)=0
R(1,185)=0
R(1,186)=0
R(1,187)=0
R(1,188)=0
R(1,189)=0
R(1,190)=0
R(1,191)=0
R(1,192)=0
R(1,193)=0
R(1,194)=0
R(1,195)=0
R(1,196)=0
R(1,197)=0
R(1,198)=0
R(1,199)=0
R(1,200)=0
R(1,201)=0
R(1,202)=0
R(1,203)=0
R(1,204)=0
R(1,205)=0
R(1,206)=0
R(1,207)=0
R(1,208)=0
R(1,209)=0
R(1,210)=0
R(1,211)=0
R(1,212)=0
R(1,213)=0
R(1,214)=0
R(1,215)=0
R(1,216)=0
R(1,217)=0
R(1,218)=0
R(1,219)=0
R(1,220)=0
R(1,221)=0
R(1,222)=0
R(1,223)=0
R(1,224)=0
R(1,225)=0
R(1,226)=0
R(1,227)=0
R(1,228)=0
R(1,229)=0
R(1,230)=0
R(1,231)=0
R(1,232)=0
R(1,233)=0
R(1,234)=0
R(1,235)=0
R(1,236)=0
R(1,237)=0
R(1,238)=0
R(1,239)=0
R(1,240)=0
R(1,241)=0
R(1,242)=0
R(1,243)=0
R(1,244)=0
R(1,245)=0
R(1,246)=0
R(1,247)=0
R(1,248)=0
R(1,249)=0
R(1,250)=0
R(1,251)=0
R(1,252)=0
R(1,253)=0
R(1,254)=0
R(1,255)=0
R(1,256)=0
R(1,257)=0
R(1,258)=0
R(1,259)=0
R(1,260)=0
R(1,261)=0
R(1,262)=0
R(1,263)=0
R(1,264)=0
R(1,265)=0
R(1,266)=0
R(1,267)=0
R(1,268)=0
R(1,269)=0
R(1,270)=0
R(1,271)=0
R(1,272)=0
R(1,273)=0
R(1,274)=0
R(1,275)=0
R(1,276)=0
R(1,277)=0
R(1,278)=0
R(1,279)=0
R(1,280)=0
R(1,281)=0
R(1,282)=0
R(1,283)=0
R(1,284)=0
R(1,285)=0
R(1,286)=0
R(1,287)=0
R(1,288)=0
R(1,289)=0
R(1,290)=0
R(1,291)=0
R(1,292)=0
R(1,293)=0
R(1,294)=0
R(1,295)=0
R(1,296)=0
R(1,297)=0
R(1,298)=0
R(1,299)=0
R(1,300)=0
R(1,301)=0
R(1,302)=0
R(1,303)=0
R(1,304)=0
R(1,305)=0
R(1,306)=0
R(1,307)=0
R(1,308)=0
R(1,309)=0
R(1,310)=0
R(1,311)=0
R(1,312)=0
R(1,313)=0
R(1,314)=0
R(1,315)=0
R(1,316)=0
R(1,317)=0
R(1,318)=0
R(1,319)=0
R(1,320)=0
R(1,321)=0
R(1,322)=0
R(1,323)=0
R(1,324)=0
R(1,325)=0
R(1,326)=0
R(1,327)=0
R(1,328)=0
R(1,329)=0
R(1,330)=0
R(1,331)=0
R(1,332)=0
R(1,333)=0
R(1,334)=0
R(1,335)=0
R(1,336)=0
R(1,337)=0
R(1,338)=0
R(1,339)=0
R(1,340)=0
R(1,341)=0
R(1,342)=0
R(1,343)=0
R(1,344)=0
R(1,345)=0
R(1,346)=0
R(1,347)=0
R(1,348)=0
R(1,349)=0
R(1,350)=0
R(1,351)=0
R(1,352)=0
R(1,353)=0
R(1,354)=0
R(1,355)=0
R(1,356)=0
R(1,357)=0
R(1,358)=0
R(1,359)=0
R(1,360)=0
R(1,361)=0
R(1,362)=0
R(1,363)=0
R(1,364)=0
R(1,365)=0
R(1,366)=0
R(1,367)=0
R(1,368)=0
R(1,369)=0
R(1,370)=0
R(1,371)=0
R(1,372)=0
R(1,373)=0
R(1,374)=0
R(1,375)=0
R(1,376)=0
R(1,377)=0
R(1,378)=0
R(1,379)=0
R(1,380)=0
R(1,381)=0
R(1,382)=0
R(1,383)=0
R(1,384)=0
R(1,385)=0
R(1,386)=0
R(1,387)=0
R(1,388)=0
R(1,389)=0
R(1,390)=0
R(1,391)=0
R(1,392)=0
R(1,393)=0
R(1,394)=0
R(1,395)=0
R(1,396)=0
R(1,397)=0
R(1,398)=0
R(1,399)=0
R(1,400)=0
R(1,401)=0
R(1,402)=0
R(1,403)=0
R(1,404)=0
R(1,405)=0
R(1,406)=0
R(1,407)=0
R(1,408)=0
R(1,409)=0
R(1,410)=0
R(1,4
```

















```

GENERAL SUBROUTINE 0040 DETERMINES THE INVERSE SIN AND
INVERSE COS OF THE PASSED ARGUMENT (PI). THE VARIABLE
P2 IS USED TO DETERMINE WHICH FUNCTION WILL BE IMPLEMENTED.
IF P2=1 THEN THE ARCSIN OF THE ARGUMENT IS DETERMINED
IF P2=2 THEN THE ARCCOS OF THE ARGUMENT IS DETERMINED
AND APPLIED TO THE POSITIVE RANGE 0-180 DEGREES.

```









```

LET PI=I6*1128
GOSUB 0030
LET C(8)=I2-PI-I3
IF C(8) GE 0 THEN 0101
LET C(8)=C(8)+C1
LET C(6)=I6-C1
LET REM CORRECT FROM 0.0 JAN 75 TO BASE DATE 29.0 NOV 71
LET C(5)=I4-C(4)*1128
RETURN
0101

```



```

** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **  **
** ** ** **  SUBROUTINE (0200) REQUEST ALL INITIAL DATA REQUIRED
** ** ** **  FOR FIX. CONSTANT DEFINITIONS FOLLOW:
** ** ** **  L2=LEAP YEAR
** ** ** **  D=DAY OF THE MONTH
** ** ** **  M=MONTH
** ** ** **  Y=YEAR (LAST TWO DIGITS 19XX )
** ** ** **  *****
** ** **  0200 LET L2=
** ** **  ***** REM SET LOGICAL FLAG 2 (LEAP YEAR) TO TRUE
** ** **  ***** REM FROM THIS POINT TO 0204 DETERMINE THE DAYS SINCE REF
** ** **  ***** REM DATE 29 NOV 1971
** ** **  ***** PRINT D
** ** **  ***** PRINT D ENTER MONTH
** ** **  ***** PRINT M
** ** **  ***** PRINT M ENTER YEAR
** ** **  ***** PRINT Y
** ** **  ***** PRINT R8=M
** ** **  ***** LET R6=30*M+D
** ** **  ***** LET R7=Y-72
** ** **  ***** LET R6=R6+365*R7
** ** **  ***** LET R7=R7/4
** ** **  ***** LET R6=R6+INT(R7)
** ** **  ***** IF (R7-INT(R7)) NE 0 THEN 0201
** ** **  ***** REM INDICATE THIS YEAR IS LEAP YEAR
** ** **  ***** LET L2=F
** ** **  ***** IF M LE 2 THEN 0203
** ** **  ***** IF M LE 8 THEN 0202
** ** **  ***** LET M=M+1
** ** **  ***** LET M=INT(M/2)
** ** **  ***** GO TO 0204
** ** **  ***** IF L2=F THEN 0204
** ** **  ***** LET M=M+1
** ** **  ***** LET R3=R6+M
** ** **  ***** PRINT A ENTER ALTITUDE (FT MSL)
** ** **  ***** READ A
** ** **  ***** PRINT A ENTER TEMPERATURE (DEG CENTIGRADE)
** ** **  ***** PRINT T1
** ** **  ***** PRINT T1
** ** **  ***** LET L1=F
** ** **  ***** PRINT L1 ENTER COURSE (D.MS)

```



```

READ P1
PRINT P1
GOSUB 0010
LET C=P1
PRINT 'ENTER SPEED (KNOTS TRUE)'
READ K
PRINT K
REM CONVERT SPEED (KTS) TO RADIAL VELOCITY
LET K=.00290882087*K
PRINT 'ENTER ASSUMED LAT (N/S) +/- (D.MS)'
READ P1
PRINT P1
GOSUB 0010
LET R1=P1
PRINT 'ENTER ASSUMED LONG(E/W) +/- (D.MS)'
READ P1
PRINT P1
GOSUB 0010
LET R2=P1
PRINT 'ENTER NUMBER OF BODIES THIS FIX'
READ N1
PRINT N1
PRINT '1 TO N1'
FOR N = 1 TO N1
  PRINT 'ENTER STAR/BODY NUMBER'
  READ R9
  PRINT R9
  IF R9 NE 44 THEN 0214
  PRINT 'CONSTANTS UNAVAILABLE'
  LET N1=N1-1
  GO TO 0209
PRINT 'ENTER GMT OF OBS (H.MS)'
0214 READ P1
PRINT P1
LET L1=1
REM CONVERT GMT (H.MMSS) TO GMT (HRS.)
GOSUB 0010
LET R4=P1
PRINT 'GMT (HRS)=',R4
IF N=1 THEN 0206
IF (O(N-1))-INT(O(N-1))) LT R4 THEN 0206
LET R3=R3+1
LET O(N)=R3+R4/24
PRINT 'ENTER HS (D.MS)'
0206 READ P1
PRINT P1
GOSUB 0010
LET H=P1

```





```

PRINT 'ENTER IC (D.MS) '
READ P1
PRINT P1
GOSUB 0010
LET I=P1
  REM COMPUTE HA=HS+IC
  LET R8=H+I
  REM *****
  REM ** CORRECT FOR ALTITUDE, TEMPERATURE AND CORIOLIS EFFECT **
  REM ** IF USED IN AIRCRAFT OR HIGH SPEED CRAFT **
  REM *****
  REM *****
  IF A GT 200 THEN 0207
  REM DETERMINE THE DIP IN MINUTES OF ARC
  LET P1=.97*SQR(A)
  REM CONVERT MINUTES OF ARC TO RADIANS
  LET P1=P1*.0002908882
  REM REMOVE THE EFFECT OF DIP FROM THE OBSERVED ALTITUDE
  LET R8=R8-P1
  LET R5=R8
  LET H(N)=R5
  LET P1=R8
  GOSUB 0020
  PRINT 'HA=',P1
  LET P1=I
  GOSUB 0020
  LET P1=H
  GOSUB 0020
  REM DETERMINE THE CONSTANTS NEEDED FOR THIS BODY
  GOSUB 0100
  REM DETERMINE GHA AND DEC FOR THIS BODY
  GOSUB 0300
  LET G(N)=P3
  LET D(N)=P4
  REM DETERMINE ZN ZND DISTANCE TOWARD THIS BODY FOR LOP
  GOSUB 0400
  NEXT N
0209 IF N1=1 THEN 0208
  REM CORRECT THE OBSERVED ALTITUDES FOR PLATFORM MOVEMENT
  GOSUB 0500
  REM DETERMINE THE FIX IF POSSIBLE
  GOSUB 0600
  LET P1=D(01)-INT(D(01))
  PRINT 'TIME OF FIX (.DAY)',P1
  RETURN

```











```
PRINT 'DISTANCE TOWARD BODY (NM)', P1
LET P1=R8
GOSUB 0020
PRINT 'BASED ON HC (D.MMSS)', P1
PRINT ' '
PRINT ' '
RETURN
```









```

** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **  SPECIAL ROUTINE 0600 RECEIVES THE GHA'S AND DEC'S FOR ALL
** ** ** **  OF THE BODIES USED IN THE STAR FIX AND DETERMINES THE THREE
** ** **  INTERSECTIONS OF THE RESPECTIVE CIRCLES OF EQUAL ALTITUDES
** ** **  THAT ARE APPLICABLE TO THE FIX (OR TWO INTERSECTIONS FOR A
** ** **  TWO BODY FIX). IN THE CASE OF THREE BODIES, THE THREE INT-
** ** **  ERSECTIONS WITH THE SMALLEST SOLID ANGLE ARE USED TO DEFINE
** ** **  THE VERTICES OF THE TRIANGLE OF WHICH THE CENTER OF MASS
** ** **  IS USED TO DEFINE THE FIX
** ** **  IN THE CASE OF A TWO BODY FIX, THE CLOSEST INTERSECTION TO
** ** **  THE ASSUMED POSITION IS THE FIX.
** ** ** **  *****
0600 IF N1 NE 2 THEN 0601
      REM WITH TWO BODIES GO TO THE INTERSECTION ROUTINE 0700
      LET I1=1
      GO TO 0602
0601 LET I1=3
      IF ONLY ONE BODY USED THEN 'FIX' NOT POSSIBLE
0602 IF N1 = 1 THEN 0613
      REM CAUSE THE INTERSECTION ROUTINE TO ITERATE ON ALL
      REM COMBINATIONS OF BODIES
      FOR Q=1 TO I1
      LET P=Z+1
      IF R NE 4 THEN 0603
      LET R=1
      LET S=2
      ***** DETERMINE THE INTERSECTIONS OF EQUAL ALTITUDES FOR THIS PAIR
0603 LET REM Z
      GOSUB 0700
      NEXT Z
      IF N1 NE 2 THEN 0605
      REM IN THE CASE OF TWO BODIES SELECT THE CLOSEST INTERSECTION
      REM AS THE FIX
      I2=SQR((L(1)-R1)**2+(E(1)-R2)**2)
      I3=SQR((L(2)-R1)**2+(E(2)-R2)**2)
      IF I2 GT I3 THEN 0604
      LET I8=L(1)
      LET I9=E(1)
      GO TO 0612
      LET I8=L(2)
      LET I9=E(2)
0604 LET TO 0612
      REM REM IN THE CASE OF THREE BODIES DETERMINE THE INTERNAL ANGLE
      REM REM BETWEEN ALL PAIRS OF INTERSECTIONS
      VI=999999
0605 LET

```



```

FOR S=1 TO 5
FOR J=(S+1) TO 6
LET P1=COS(ABS(L(S)-L(J)))*COS(ABS(E(S)-E(J)))
LET P2=2
REM REM DETERMINE THE ARC COS OF THE INTERIOR ANGLES
COSUB 0040
LET RO=P1
REM REM SELECT THE SMALLEST INTERIOR ANGLE
IF VI LT RO THEN 0606
REM REM STORE THE INTERIOR ANGLE AND THE PARTICIPATING BODIES
LET VI=RO
LET MI=S
LET M2=J
LET V(S,J)=RO
NEXT J
NEXT S
0605
REM REM DETERMINE THE SMALLEST SOLID ANGLE OF THE TRIANGLES THAT
REM REM INCLUDE THE TWO INTERSECTIONS THAT PRODUCE THE SMALLEST
REM REM INTERIOR ANGLE
FOR Q=1 TO 6
IF Q GE MI THEN 0607
LET DI=V(Q,MI)+V(Q,M2)
GO TO 0609
IF Q GE M2 THEN 0608
IF Q GE MI THEN 0510
LET DI=V(M1,Q)+V(Q,M2)
GO TO 0609
IF Q = M2 THEN 0510
IF DI=V(M1,Q)+V(M2,Q)
LET V(R+1,I)=DI
LET V(6,R)=Q
LET R=R+1
NEXT Q
LET DI=99999
FOR Q=2 TO 5
IF V(Q,I) GT DI THEN 0611
LET DI=V(Q,I)
LET R=Q-1
NEXT Q
0611
REM REM DETERMINE THE CENTER OF MASS OF THE TRIANGLE USING
REM REM THE THREE SELECTED SIDES
I1=V(6,R)
I2=M1
I3=M2
I4=L(I1)-L(I2))/3
I5=E(I1)-E(I2))/3
I6=L(I1)-L(I3))/3

```



```

LET I7=E(I1)-2*(E(I1)-E(I3))/3
LET I8=I4+(I6-I4)/2
LET I9=I5+(I7-I5)/2
LET PI=I8
GOSUB 0020
LET I8=PI
LET PI=I9
GOSUB 0020
LET I9=PI
0612 PRINT 'FIX (LAT/LONG D.MMSS)=',
0613 RETURN

```









```

      P2=2
      GOSUB 0040
      IF L2=F THEN 0705
      LET P1=-P1
0705  LET E(J6)=P1-G(R)
      LET J=1
      IF ABS(E(J6)) LT C4 THEN 0707
      IF E(J6) GT 0 THEN 0706
      LET J=-1
0706  LET E(J6)=E(J6)-2*C4*J
0707  NEXT I
0708  RETURN

```

END



# APPENDIX B SAMPLE PROGRAM RUN

```

ENTER DAY
20
ENTER MONTH
6
ENTER YEAR
75
ENTER ALTITUDE (FT MSL)
60
ENTER COURSE (D.MS)
270
ENTER SPEED (KNOTS TRUE)
5
ENTER ASSUMED LAT (N/S) +/- (D.MS)
40
ENTER ASSUMED LONG (E/W) +/- (D.MS)
15.2224
ENTER NUMBER OF BODIES THIS FIX
3

ENTER STAR/BODY NUMBER
1
ENTER GMT OF OBS (H.MS)
.56
GMT (HRS)= .944443
ENTER HS (D.MS)
37.04
ENTER IC (D.MS)
0
HA= 36.5709
GHA OF BODY= 280.043
DEC OF BODY (D.MMSS) 28.5728
ZN OF BODY (D.MMSS) 80.4231
DISTANCE TOWARD BODY (NM) 8.25071
BASED ON HC (D.MMSS) 36.4854

ENTER STAR/BODY NUMBER
46
ENTER GMT OF OBS (H.MS)
1
GMT (HRS)= 1
ENTER HS (D.MS)
48.39
ENTER IC (D.MS)
0
HA= 48.3209
GHA OF BODY= 19.1425
DEC OF BODY (D.MMSS) 12.3436
ZN OF BODY (D.MMSS) 237.462
DISTANCE TOWARD BODY (NM) -30.9003
BASED ON HC (D.MMSS) 49.0303

ENTER STAR/BODY NUMBER
40
ENTER GMT OF OBS (H.MS)
1.04
GMT (HRS) 1.07778
ENTER HS (D.MS)
42.11
ENTER IC (D.MS)
0
HA= 42.0409
GHA OF BODY= 61.1234
DEC OF BODY (D.MMSS) 74.1533
ZN OF BODY (D.MMSS) 339.153
DISTANCE TOWARD BODY (NM) 14.4483
BASED ON HC (D.MMSS) 41.4942

```



ENTER SHOT OF DESIRED FIX TIME

<sup>2</sup>  
CORRECTED DISTANCE TOWARD BODY 1

7.92129  
BASED ON CORRECTED HO (D.MMSS)  
36.4834

CORRECTED DISTANCE TOWARD BODY 2

-30.9003  
BASED ON CORRECTED HO (D.MMSS)  
49.0303

CORRECTED DISTANCE TOWARD BODY 3

14.3302  
BASED ON CORRECTED HO (D.MMSS)  
42.0402

FIX (LAT/LONG D.MMSS)=  
40.3313 15.3354

TIME OF FIX (.DAY)=

.0415566

PROGRAM COMPLETE





## BIBLIOGRAPHY

1. Weir, Joseph, Interactive Information Systems, Bell Telephone Laboratories. Hayden Book Co., Inc., Rochelle Park, New Jersey, 1974.
2. Peebles, Richard W., "Design Considerations for Distributed Data Access Systems," Technical Report, University of Pennsylvania.
3. Electronics, "C-LAD Steers Ship by Computer," April 18, 1974, p. 31.
4. Electronics, "Low Cost LORAN, Transit Developed," Sept. 11, 1972, p. 32.
5. Rudnick, Steve, "Microprocessors Unmasked," Digital Design, June 1974, Benwill Publishing Corp.
6. Wolf, Helmut F., "Microcomputers-Revolution or Evolution," Digital Design, June 1974, Benwill Publishing Corp.
7. Dejka, W. J., "Navy Opportunities for Microcomputers," Nav Sea Journal, Jan 1975, Published by Naval Sea Systems Command.
8. Nuese, Carl and Schneider, Dr. Alan, "Celestial Navigation Using a Hand-held Programmable Calculator," NELC Technical Note 2873, 22 Jan 1975 (A tentative and unpublished report).
9. Fox, E. J., "Proceedings of the Symposium on Computer Communications Networks and Teletraffic," Polytechnic Institute of Brooklyn, C. R. 1972.
10. The American Ephemeris and Nautical Almanac published annually in Washington by the Naval Observatory and in London by Her Majesty's Nautical Almanac Office under the title The Astronomical Ephemeris, viii, 568 pages.
11. Explanatory Supplement to the Astronomical Ephemeris prepared jointly by the Nautical Almanac Offices of the United Kingdom and the United States of America.
12. Intel Corporation, "A Guide to PLM Programming," Rev. 1, Santa Clara, CA, 1973.
13. The Nautical Almanac published annually in Washington by the United States Naval Observatory and in London by Her Majesty's Stationery Office, xxxvi, A4, 276 pages.



14. Fricke, W., and Lederle, T., Apparent Places of Fundamental Stars 1975, Heidelberg, Astronomisches Rechen-Institut, 1973, xiiv, 510 pages.
15. Digital Design, Dec 1974, Benwill Publishing Corp., p. 8.



# INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Chairman, Computer Science Group, Code 72 Naval Postgraduate School Monterey, California 93940	1
4. Asst Professor V. M. Powers, Code 72 Pw Computer Science Group Naval Postgraduate School Monterey, California 93940	3
5. Asst Professor G. A. Kildall, Code 72 Kd Computer Science Group Naval Postgraduate School Monterey, California 93940	1
6. LTJG G. M. Raetz, Code 72 Rz Computer Science Group Naval Postgraduate School Monterey, California 93940	1
7. LT Jon P. Moore, USN 771 Ladysmith Drive El Cajon, California 92020	1
8. LT David R. Rainsberger, USN 113 Coman Street Clinton, Michigan 49236	1



14 FEB 77  
17 AUG 77  
2 FEB 79

24252  
24947  
24428

161025

Thesis  
M802  
c.2

Moore

The design of a  
celestial navigation  
microcomputer with  
thoughts on an integra-  
ted information distri-  
bution system.

14 FEB 77 JAN 12 '78  
17 AUG 77  
2 FEB 79

24252  
24947  
24428

161025

Thesis  
M802  
c.2

Moore

The design of a  
celestial navigation  
microcomputer with  
thoughts on an integra-  
ted information distri-  
bution system.

thesM802

The design of a celestial navigation mic



3 2768 002 04763 1

DUDLEY KNOX LIBRARY

C.1